Dynamic Bluetooth Beacons for People with Disabilities

Alexandre Alapetite
Alexandra Institute
Copenhagen, Denmark
alexandre.alapetite@alexandra.dk

John Paulin Hansen
Technical University of Denmark
Kongens Lyngby, Denmark
jpha@dtu.dk

***Abstract*—This paper focuses on digital aids for sight impairment and motor disabilities. We propose an Internet of Things (IoT) platform for discovering nearby items, getting their status, and interacting with them by e.g. voice commands or gaze gestures. The technology is based on Bluetooth Low Energy, which is included in consumer electronics such as smartphones without requiring additional hardware. The paper presents a prototype platform illustrated by concepts of use.**

***Index Terms*—Bluetooth, beacons, interaction, assisted living, navigation, handicap.**

I. Introduction

With the advent of Bluetooth Low Energy (BLE) in smartphones (~2011), Bluetooth beacon concepts such as Apple iBeacon and Google Eddystone have emerged as a cheap and efficient way to support indoor localization and navigation. However, such basic beacons typically broadcast static information, and neither iBeacon nor Eddystone natively accept user input.

Common Bluetooth beacons are subsets of the standard Bluetooth Low Energy two-way communication protocol. In this paper, we propose to extend the basic Bluetooth beacons to make them broadcast a dynamic payload, to convey their state (e.g. on / off, in use / available…). Finally, we propose to configure our dynamic beacons to accept user inputs (e.g. turn on / off).

Adding the concept of dynamic payloads and actuation to normal BLE beacons makes it possible for a simple IoT platform to discover things, get their status, and actuate them, all in the same workflow.

II. Background

Indoor positioning is a key enabling technology for IoT applications [1], and smart products in particular.

GPS provides a great service, particularly for outdoor navigation when combined with a compass, especially now that both are commonplace in smartphones. However, GPS is not accurate enough for indoor guiding. There has been considerable research on indoor navigation [1, 2], often combining multiple technologies [2] such as Wi-Fi positioning, active RFID, ultrasound, etc. However, despite years of availability, such technologies have not been broadly deployed so far, probably due to cost and scalability issues in particular. Bluetooth beacons are cheap, relatively easy to deploy, and can interact with smartphones without requiring any additional hardware. Bluetooth beacons typically broadcast a short message (~30 bytes), and allow the smartphone to compute an estimation of the distance to the beacon, based on the radio signal strength (RSSI) [3]. Those assets are some of the reasons for the current popularity of Bluetooth beacons.

Finally, we believe it is desirable to combine indoor navigation with human-machine interaction possibilities [4], all within the same platform.

III. Use cases

The system presented in this paper has been developed with two use-cases in mind: people challenged by sight impairment, or by motor disability. Ultimately, end-users will be served with distinct, customised user interfaces, but both groups will be applying “proxemic” interaction [4] where dynamic beacons mediate their control of near devices.

A. Sight impairment

People with vision disability have great difficulties navigating both indoors and outdoors environments. Guide dogs are used by some, but they are not sufficient during novel situations, such as when shopping in an unfamiliar area. Together with the Institute for the Blind and Partially Sighted (Denmark), we observed that a majority of their members actively use a modern smartphone (with “voice over” assistive technology). They repeatedly expressed reluctance to buy and carry any additional equipment. Dedicated electronic assistive devices are thus not widely used. Making applications based solely on users’ existing smartphones is therefore crucial.

Using Bluetooth beacons such as Apple iBeacon is already a promising approach for the vision disabled [5], but we push the concept further by exploring the potentials of dynamic beacons, i.e. beacons with the ability to get a dynamic state and to actuate.

B. Physical impairment

The focus for physical impairment is on people moving around in a future smart-home on a wheelchair. When approaching a door, for instance, a smartphone (or similar device) mounted on the wheelchair will sense the proximity to the door and suggest opening it, c.f. Fig. 1:

978-1-5090-4130-5/16/$31.00 ©2016 European Union

Fig. 1. Use-case illustration: Opening a door by proximity discovery of a Bluetooth-enabled door actuator and executed by the user's gaze gestures detected with a pair of JINS MEME eye-tracking glasses.

The user interface could then allow a gaze-gesture command [6] (provided that the smartphone – or some device associated with it – can track the eyes) or a speech command (provided that the user can speak). If several interaction possibilities are present at the current location, they are presented in a list. For instance, controlling the light when a TV-monitor and door are nearby and thus appearing on the list may be done by gaze gestures to select the object to control (i.e. the light). A gesture to the right would bring up two possibilities for this object (i.e. increase or decrease), and a final gesture upwards would then increase the luminosity (c.f. Fig. 3).

IV. PROTOTYPING PLATFORM

Two different hardware platforms were selected to implement the custom dynamic Bluetooth beacons. Both platforms are relatively cheap, well-known, and easy to build upon. While the two platforms have some overlapping use cases, they have their distinctive qualities.

A. RFduino (microcontroller)

Largely compatible with the well-known Arduino platform, RFduino[1] can be programmed over USB with an Arduino IDE. It is a microcontroller (32-bit 16MHz ARM) with built-in Bluetooth 4.0 LE, and libraries for programming it as a beacon. We use it for making custom beacons that need to run on battery, or only have low software requirements (Fig. 2; Fig. 3).

B. Raspberry Pi (small computer)

A microcontroller like an RFduino is not very convenient for prototyping multimedia applications, for interacting with USB devices, for Internet connectivity, and for running heavier software. In these cases, a small computer is more appropriate (Fig. 4). We use Raspberry Pi[2], the most popular credit-card size computer for IoT. It runs a full Linux OS. The high-end Raspberry Pi 3 Model B has built-in Bluetooth connectivity, but the much cheaper Raspberry Pi B+ or even A+ can use an inexpensive USB Bluetooth dongle and consume less energy.

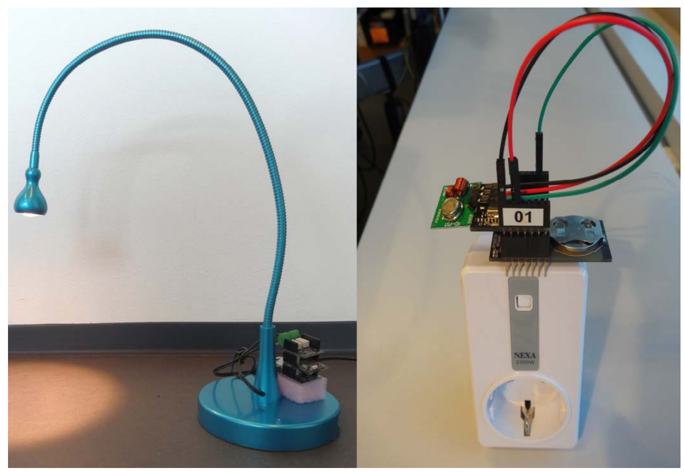

Fig. 2. 4V lamp controlled by an RFduino relay (left). Remote-controlled plug controlled by an RFduino and a 433MHz radio module (right)

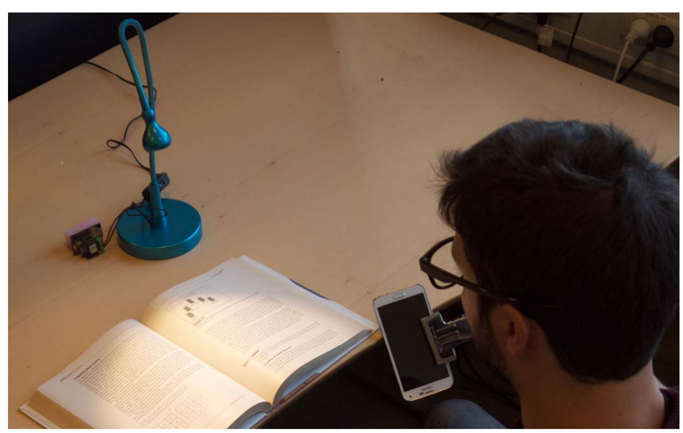

Fig. 3. A user controlling the IoT lamp by gaze gestures.

[1] http://www.rfduino.com

[2] https://www.raspberrypi.org

Fig. 4. Raspberry Pi with an infrared emitter (left).
Raspberry Pi with an USB audio speaker (right).

V. Examples of dynamic beacons

This section describes several concepts of dynamic beacons.

A. Electric relay

We equipped several LED lamps with an electric relay controlled by an RFduino (Fig. 2, left). The modified lamp broadcasts its status (on/off), can be discovered, and accepts commands (on/off) by Bluetooth (Fig. 3).

B. Remote-controlled electric plug

For higher voltages than the LED lamp, we use an off-the-shelf radio-controlled plug (Fig. 2, right), for which an RFduino beacon replaces the remote control (using a 433MHz module). With this approach, we keep the integrity of the electric plug and its certification. This beacon also broadcasts the state of the plug (on/off), can be discovered, and accepts commands (on/off) by Bluetooth. For instance, we may use this system for Bluetooth-enabling larger lamps and air fans. In the case of people with low vision, it allows finding and getting the state of the coffee machine without having to touch it by hand and potentially getting injured.

C. Infrared controller

A number of domestic appliances such as televisions use infrared for communication. To make a television appear as a dynamic beacon in our platform (Fig. 4, left), we equip it with a Raspberry Pi (could also be done with an RFduino), which receives users' commands by Bluetooth, and transform them into the corresponding infrared signals. In the simplest version, the modified TV broadcasts its status (on / off), can be discovered, and accepts commands (on / off) by Bluetooth.

D. Audio beacon

Our audio beacon (Fig. 4, right) can be used to play music on demand, and it is especially useful for playing some brief and easily identifiable sounds to guide the navigation of people with vision impairment. People can discover audio beacons in their vicinity from their smartphone, and activate them one at a time. For instance, with audio beacons on the fridge, the toilets, and the exit door, users are able to trigger a sound on any of them, and easily walk towards the points of interest guided by their hearing.

Furthermore, it is possible to have a series of audio beacons on a longer indoor walk, using them as waypoints, for instance down a hallway.

E. Other examples

These few examples of dynamic Bluetooth beacons can be extended to several electronic devices. In particular, both the RFduino and Raspberry Pi-based beacons can communicate with existing electronic systems via e.g. I^2C. For instance, we plan to equip an adjustable electric bed to be controlled by Bluetooth.

Finally, it is possible for a beacon to combine multiple roles, e.g. audio beacon and electric plug at the same time.

VI. Communication protocol

When broadcasting their status, the dynamic beacons can be discovered at the same time by many client devices. This scales well. However, the size of the broadcasted payload is limited to a few bytes (as of Bluetooth version 4). The current payload broadcasted by our dynamic beacons has the following format: "`DynBeacon01=XY`", where "`DynBeacon`" is a fixed string to distinguish different families of beacons, "`01`" is an identifier for the beacon, "`X`" is the type of beacon (e.g. "lamp"), and "`Y`" is the status (e.g. "`1`" for "on").

Only one client can send a command to the beacon at a time, but the process takes just a few milliseconds. To send a command, the client opens a Bluetooth GATT connection, and sends the desired state (e.g. "`1`" for "switch on"). This command can be accompanied by a password if it is preferable to only allow commands from selected clients and for additional security.

VII. Client applications

Requiring users to install an app is often a barrier to adoption. Currently, the discovery part can be done without an app on some smartphone models. For instance, it is possible to have Bluetooth beacon notifications on Android devices without having to install an additional app, thanks to the Google Eddystone beacon format[3], which is built into Google Chrome (since version 49; often pre-installed on Android devices). However, such automatic notifications are not as instantaneous and reliable as when a dedicated app actively scans for beacons of interest in the vicinity. This is the same problem when using background iBeacon scanning on Apple iOS. In the near future, when the upcoming HTML5 Web Bluetooth API[4] is implemented by mobile devices, it will be possible to have a Web client for an IoT platform, which is so far only possible if connecting the beacons to Internet (which can easily be done with beacons based on Raspberry Pi) for receiving user commands by Internet instead of by local Bluetooth.

---

3 https://github.com/google/eddystone

4 https://webbluetoothcg.github.io/web-bluetooth/

For the time being, our early prototypes are an Android app (Fig. 5) (an iOS version would be very similar) as well as a Microsoft Windows version (Universal Windows) (Fig. 6), which are doing active scanning of beacons, retrieve their type (e.g. “lamp”) and status (e.g. “off”), their identifier (especially in the case when multiple beacons of the same type are nearby), and offer a user interface to actuate selected beacons (e.g. “switch lamp on”).

On the Android app (Fig. 5), one can see a list of beacons in the vicinity, with their name, their type (conveyed by text and icon), and their status (conveyed by text and colour). Clicking on the icon toggles the status (on / off) of the selected beacon.

On the Windows app (Fig. 6), one can see a list of beacons in the vicinity, sorted by estimated proximity, with their name and type conveyed by text, and their status conveyed by text and colour. Clicking on a beacon’s line toggles the beacon’s status (on / off). The user interface also reacts to voice commands (e.g. “switch on”, “switch off”, which sends the appropriate command to the nearest beacon) as well as eye gestures (lateral gesture for “off”, vertical gesture for “on”). The Windows platform is currently limited by the fact that Bluetooth beacons must be manually paired to the client device prior to first-time use.

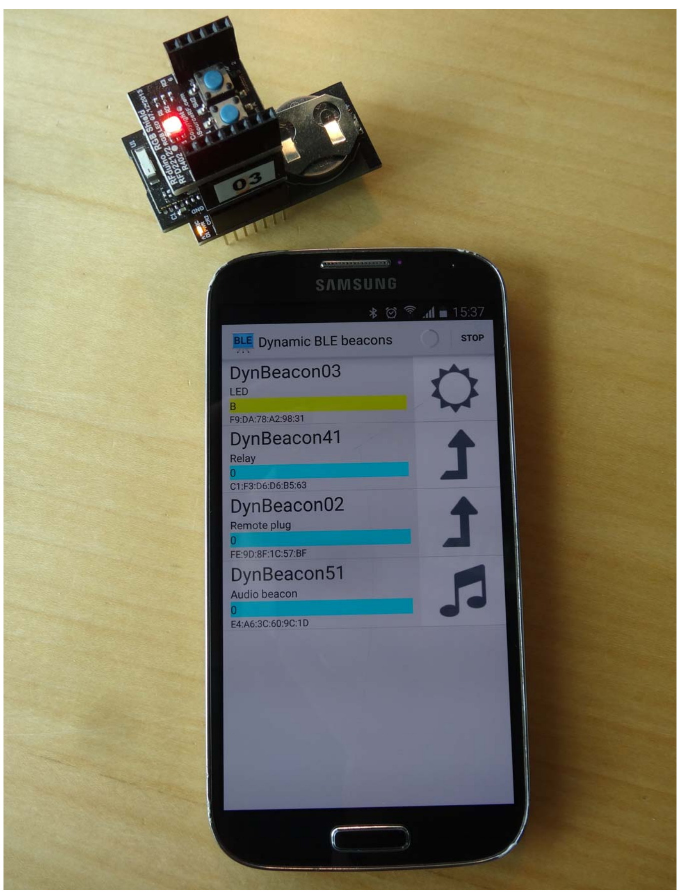

Fig. 5. Google Android app prototype,
next to a basic Dynamic Beacon with a LED switched on
(which appears in yellow at the top of the list on the smartphone)

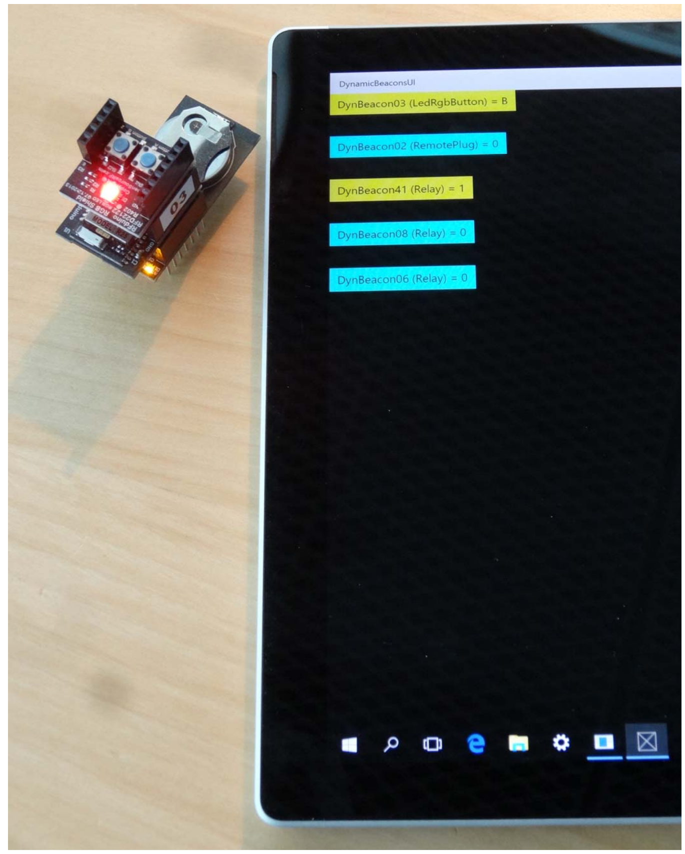

Fig. 6. Microsoft Windows prototype,
next to a basic Dynamic Beacon with a LED switched on
(which appears in yellow at the top of the list on the tablet)

VIII. Directional Information

While this platform offers information regarding relative proximity between the user (smartphone) and the nearby devices equipped with a beacon, there is no information about the direction (e.g. left / right) the user should head to in order to reach the beacon. In other words, the platform does not know whether the user is facing a beacon, or turning his/her back to it. Indeed, Bluetooth antennas used in current smartphones and common beacons are omnidirectional, and therefore do not natively provide direction information.

Different workarounds are possible, e.g. by using prior knowledge of the location of the beacons combined with indoor navigation techniques such as taking advantage of the compass sensor built in smartphones as well as the user’s recent movements.

Another approach, more unique to Bluetooth beacons, is to make multiple signal strength measurements from the smartphone when the user rotates on her/himself, and using the fact that the human body is a good shield for 2.4GHz radio signals used by Bluetooth LE. Therefore, the signal strength is significantly lower when the beacon is located precisely in the back of the user (with the smartphone in front of the chest), compared to when the user is more or less facing the beacon.

Regarding the user interface, we are currently investigating prompting the user to look at, or point towards, the object he/she would like to interact with and then track this movement from the mobile device [6]. This additional action may only be needed when several objects are found within the pre-set proximity distance.

IX. IoT Security

It is a challenge to properly balance IT security with functionalities, ease of use, hardware limitations, and energy constraints, in particular. The security of IoT networks based on Bluetooth Low Energy has been studied, reporting a number of threats [7].

In our more precise case: while classic Bluetooth systems often use IT security protections such as pairing, common Bluetooth beacons simply broadcast some information without any cryptographic guarantee. It makes Bluetooth beacons in particular vulnerable to cloning, i.e. someone can easily replicate the same beacon signal from another location. There are various approaches to mitigate the issue, such as falling back to Bluetooth pairing at least for actuations, or having the client application open a Bluetooth connection to the beacon to perform a cryptographic challenge. Furthermore, there are a number of additional IoT security challenges, for instance a vulnerability to denial of service if another client permanently attempts to open a connection to the beacon.

We recognize that the IoT security is a major point to improve prior to a larger use of Bluetooth beacons, and dynamic beacons in particular. At the current stage, this is what we believe is the main drawback of the proposed approach.

X. Performances

From the user's perspective, the detection of the dynamic beacons in the vicinity is almost instantaneous. The detection of a state change is just as fast. Indeed, our beacons are programmed to broadcast their dynamic payload 10 times per second.

During a light performance testing at a distance of 1 meter, it was possible to detect our IoT lamp, receive its status (on / off), and perform a click to change the status, 26 times in average for a 30s run (tested with the Android app), which is almost one full cycle per second.

Farther away from the beacon, there is a higher likelihood of losing some packets, which might add some latency to the detections. However, thanks to the high broadcast frequency, we could not observe any significant impact even 20 meters away. Adding additional security layers would decrease the performances, in particular for actuation.

XI. Discussion and conclusion

For an IoT platform to gain momentum within assistive technologies and other areas, it should ideally be applicable to multiple use cases. That is the reason why we are targeting disabilities at large, with an eye on users without handicap for future developments. Though not developed in this article, we consider other use cases, such as people with cognitive disabilities [8], who can get location-specific help (like the NemInfo system[5]) as well as custom user interfaces on their smartphone to interact with devices in their environment. People without disabilities could also become interested in the platform, especially in a smart house context, conferences, shopping experiences, etc.

Furthermore, to gain adoption, the same technological base should be used in multiple places (as opposed to e.g. one single building), and we believe this can best be done with a decentralised model (no central actor) and by allowing custom user interfaces on the client's side. For that, a very promising concept is to push a user interface from the beacons themselves onto the user's smartphone, like Simblee[6] does it. This concept removes the need for installing many custom apps, but requires a standardisation prior to broad acceptance.

Finally, the upcoming Bluetooth 5 standard will allow some improvements to the concept of dynamic beacons, in particular thanks to the increased broadcasting capacity with a larger payload.

References

[1] Davide Macagnano, Giuseppe Destino, Giuseppe Abreu. "Indoor positioning: A key enabling technology for IoT applications". IEEE World Forum on Internet of Things (WF-IoT 2014), 6-8 March 2014, pp 117-118, doi:10.1109/WF-IoT.2014.6803131

[2] Kari Rye Schougaard, Kaj Grønbæk, Tejs Scharling. "Indoor Pedestrian Navigation Based on Hybrid Route Planning and Location Modeling". Proceedings of the 10th International Conference, Pervasive 2012, Newcastle, UK, June 18-22, 2012, doi:10.1007/978-3-642-31205-2_18

[3] Joonyoung Jung, Dong-oh Kang, Changseok Bae. "Peer to peer signal strength characteristic between IoT devices for distance estimation". IEEE World Forum on Internet of Things (WF-IoT 2014), 6-8 March 2014, Seoul, pp 208 - 211, doi:10.1109/WF-IoT.2014.6803160

[4] Till Ballendat, Nicolai Marquardt, Saul Greenberg. "Proxemic interaction: designing for a proximity and orientation-aware environment". In ACM International Conference on Interactive Tabletops and Surfaces (ITS '10). ACM, New York, NY, USA, 121-130. doi:10.1145/1936652.1936676

[5] A. Joseph Ruffa, Amy Stevens, Nicholas Woodward, Torin Zonfrelli. "Assessing iBeacons as an Assistive Tool for Blind People in Denmark". Worcester Polytechnic Institute, Interactive Qualifying Project E-project-050115-131140, May 2015

[6] John Paulin Hansen, Haakon Lund, Florian Biermann, Emillie Møllenbach, Sebastian Sztuk, Javier San Agustin. "Wrist-worn pervasive gaze interaction". In Proceedings of the 9th Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA'16). ACM, New York, NY, USA, 57-64. doi:10.1145/2857491.2857514

[5] Living IT Lab, NemInfo http://www.neminfo.net

[6] RF Digital Corporation, Simblee https://www.simblee.com

[7] Yanzhen Qu, Philip Chan. "Assessing Vulnerabilities in Bluetooth Low Energy (BLE) Wireless Network Based IoT Systems". IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), 9-10 April 2016, New York, NY, USA, pp 42-48, doi:10.1109/BigDataSecurity-HPSC-IDS.2016.63

[8] Yao-Jen Chang, Yen-Yin Chu, Chen-Nien Chen, Tsen-Yang Wang. "Mobile Computing for Indoor Wayfinding Based on Bluetooth Sensors for Individuals with Cognitive Impairments". 3rd International Symposium on Wireless Pervasive Computing (ISWPC 2008), 7-9 May 2008, Santorini, IEEE, pp 623-627, doi:10.1109/ISWPC.2008.4556284